Novel Approaches to Identify Distracted Drivers

Project Report - DATA7703

THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

Hinton's Disciples

Lalith Veerabhadrappa Badiger (S4655782) | Daniel Herr (S44556994)

Dong Hee Lee (S4247012) | Siddharth Krishnakumar (S4651743)

We give consent for this to be used as a teaching resource.

Table of Contents

Abstract

Distracted driving describes the act of driving while engaged in any activity that draws drivers' attention away from the road. Identifying distracted driving is important because they are much more likely to be involved in an accident, and they are a threat to other road users.

In this project, we approached the task of autonomous detection of distracted drivers by processing image data with a variety of techniques from the discipline of deep learning (i.e. Convolutional Neural Networks, Transfer Learning) and built a classification model with a validation accuracy of 85.71% and a test accuracy of 62.17%.

Ultimately, we present the case that our research is a proof of concept for using this technology to be used for detecting distracted drivers in a real-world setting.

Introduction

Technological developments tend to race ahead of constitutional legal frameworks, and even once the law catches up, the practice of enforcing laws trails behind. For instance, mobile phones were introduced into Australia in 1987, and data quickly accumulated that supported a ban on using phones while driving, it only became illegal as late as 2010 in some parts of the country (Transport Operations Act *1995*). Following the introduction of these new laws, the Department of Transport and Main Roads relied upon the police to physically catch people violating these rules, but this is changing.

This project aims to aid government efforts to curtail preventable road accidents by building predictive models to enhance the capabilities of newly implemented driver monitoring technology, with novel methodology from the disciplines of computer vision and deep learning.

The dataset to be used has been purpose-built and was published to Kaggle by State Farm Insurance. It contains images of drivers seated at the steering wheel of a car, exhibiting various behaviors, some of which are negligent and dangerous (State Farm Distracted Driver Detection | Kaggle, 201). We grouped images of the dangerous activities into a set labelled 'distracted' and subsequently attempted to build a model capable of autonomously identifying reckless behavior.

The report begins by outlining the significance of the research and the potential applications for this technology. Then the available data is explored with a focus on describing the notable data characteristics. This is followed by an overview of the modeling approaches used and a description of their performance. Furthermore, we encountered severe limitations in terms of data acquisition, and these issues are described before the report concludes.

Significance

Identifying distracted driving is essential because distracted drivers are a danger on the roads as they are much more likely to be involved in an accident (Monash University Accident Research Centre, 2003; Stavrinos et al., 2013). For instance, it's estimated that approximately 12% of US road fatalities are a direct result of distracted driving and that drivers are 23 times more likely to crash while texting (Wilson & Stimpson, 2010). Moreover, the issue of distracted driving has become acute in recent years with the surge in mobile phone usage as local road fatalities are highly correlated with the volume of text message output from the region (Wilson & Stimpson, 2010).

As of 26th July 2021, Queensland introduced some of the most severe penalties for using or touching a mobile phone while driving in Australia (Transport Operations Act *1995*). The introduction of these penalties follows an announcement from the Queensland Department of Transport and Main Roads (2021) that distracted drivers caused an annual average of 29 deaths and 1284 hospitalizations in Queensland between 2015 and 2019. Developing an automated deep learning approach capable of processing images could help enhance the capabilities of the existing newly implemented driver monitoring technology. Additionally, it could also influence future hardware installation decisions. In the next section, we outline how we approached this task.

Potential Applications

The output of this project could enhance the existing systems implemented by the Department of Transport and Main Roads QLD. More precisely, this research could make roadside driver monitoring cameras more accurate, identify additional cases of driver distraction, and make the task of driver monitoring less labor-intensive.

Additionally, this technology could be used to enhance physical police surveillance and compliance by powering discrete image processing units fitted to the vehicles (or even helmets) of law enforcers. This implementation of our technology could help police drivers remain focused on their driving and collect evidence of misconduct at the same time. A benefit of this technology is that a trained model can function offline and process photos without storing them for human observation, so this point may address privacy and ethical considerations that one may anticipate.

There are also many futuristic, and perhaps, controversial applications of our project within the commercial transit and insurance sectors, in which the driving behavior of employees, or customers, could be autonomously monitored to determine suggested commercial relationships.

Feasibility

Distracted driving describes the act of driving while engaged in any activity that draws drivers' attention away from the road. Driver distraction can loosely be grouped into 3 categories; visual, cognitive, and physical (Young & Regan, 2007). Visual distractions cause drivers to take their eyes off the road, and cognitive distractions are internal and take the driver's mind away from the task of driving altogether (Papantoniou, Papadimitriou, & Yannis, 2017). These types of distractions are more difficult to detect than physical distractions that cause drivers to remove their hands from the wheel.

While it may be difficult to detect certain forms of cognitive driver distraction such as fatigue or absent-mindedness, the other forms of driver distraction, like texting on a mobile phone, are more explicit and therefore easier to detect, as shown in figure 1.

Figure 1. Sample images of distracted drivers.

The driver distraction in figure 1 is very explicit. The variety of distracting tasks is exhaustive, but mobile phone usage is a particular problem as it represents a visual, physical, and cognitive distraction.

It seems reasonable to expect that deep learning methodology will distinguish these types of activities from drivers that are not distracted, as shown in figure 2.

Figure 2. Example images of focused drivers.

The drivers in figure 2 are not explicitly distracted, and they present very differently from drivers in the preceding figure. While it may be possible to ascertain some measure of cognitive disengagement from these photos, our study is not directly concerned with that type of distraction (for the time being).

Data Specification

The dataset to be used contains over 22,000 labelled colour photos of 81 different individuals driving a variety of vehicles, performing 10 distinct tasks. We decided to approach this as a task of binary classification, so we assigned each task to one of two categories: distracted or focused. On the one hand, tasks allocated to the distracted category are using one's phone (texting or talking) and reaching for items on the back seat. On the other hand, the focused category contains images of classic safe-driving, operating the radio, drinking water, adjusting hair/makeup, and talking to a passenger.

Data Pre-processing

To enable a model to train efficiently and within a reasonable timeframe, input data should not be of an excessive magnitude. Unfortunately, each of the images in this dataset was quite large as they are of a high resolution 640 x 480 x 3 (height x width x depth) and approximately 50kb in size. Therefore, it was essential to reduce the image resolution to train our models more easily, but not to the extent that it prohibited the generation of capable models. We lowered the resolution of the images to a resolution of 224 x 224 x 3 as this retained a level of image detail that appeared to represent a preferable balance of training speed and accuracy. Additionally, as colour images are representations of 3D arrays, we considered converting the images to grayscale to reduce the dimensionality of the training dataset, but this was not suitable for the modeling approach that we adopted (transfer learning). This is because the ImageNet Dataset (used to train the models we later implement) consisted exclusively of coloured images, so converting the training data to grayscale would be inconsistent.

The pre-processing alterations significantly reduced the size of the training images, which made it a lot faster to train the model and observe how the models responded to hyperparameter adjustments.

The dataset consists of 10 folders, namely, “C0” to “C9”, where each folder represents different driver behavior. As this is a binary classification problem, these 10 folders were divided and merged into just two folders labelled: “distracted drivers” and “focused drivers”. Following the data pre-processing stage, we experimented with several Deep Learning approaches.

Modeling

The deep learning approach that we used to address this problem is Convolutional Neural Networks (CNNs). CNNs are deep learning algorithms designed for processing image data, and they are well suited for handling this task.

Initially, we implemented a vanilla CNN model that can be subsequently adjusted. The vanilla CNN model consisted of a network of 4 strided CNN layers, 1 Flatten layer, and 2 Dense layers.

This basic model and its data input were then be modified in the following ways:

- Applied batch normalization, max pooling, and dropout layers.
- Used data augmentation methods to make the model translational invariant by generating new training data with Keras "ImageDataGenerator".
- Applying Transfer Learning methods (i.e. VGG16 and ResNet) to reduce training time without sacrificing accuracy

Ultimately this is a problem of binary classification, so we could use either binary cross-entropy or categorical cross-entropy for error calculation. We decided to use categorical cross-entropy because it is a more general loss function, whereas binary cross-entropy is strictly restricted to binary classification (i.e. binary cross-entropy loss is a special case of categorical cross-entropy loss for no. of class = 2). The output of categorical cross-entropy is a probability vector that gets converted to a one-hot encoded array by finding the class of the highest probability.

Moreover, the most appropriate optimizer for our transfer learning models and vanilla CNN model is the 'Adam' optimizer which is a combination of two gradient descent methodologies: Momentum and Root Mean Square Prop (or RMSProp). The loss function used is categorical cross-entropy loss given by equation 1:

True probability distribution
(one-hot)

$$H(p,q) = -\sum_{x \in \text{classes}} p(x) \log q(x)$$

Your model's predicted
probability distribution

Equation 1. The formula for categorical cross-entropy.

Next, we outline how this technology could be applied in the case of Transfer Learning Models (VGG16 and ResNet), along with the Vanilla CNN Model implementation to classify distracted drivers.

Transfer Learning

Transfer learning effectively recycles knowledge learned from one problem and applies it to another. This process is analogous to a model trained to identify violins, adapting some of this knowledge to recognize other musical instruments like cellos or violas. It follows that transfer learning enhances efficiency in the model training process. The transfer learning process is presented in figure 3 below.

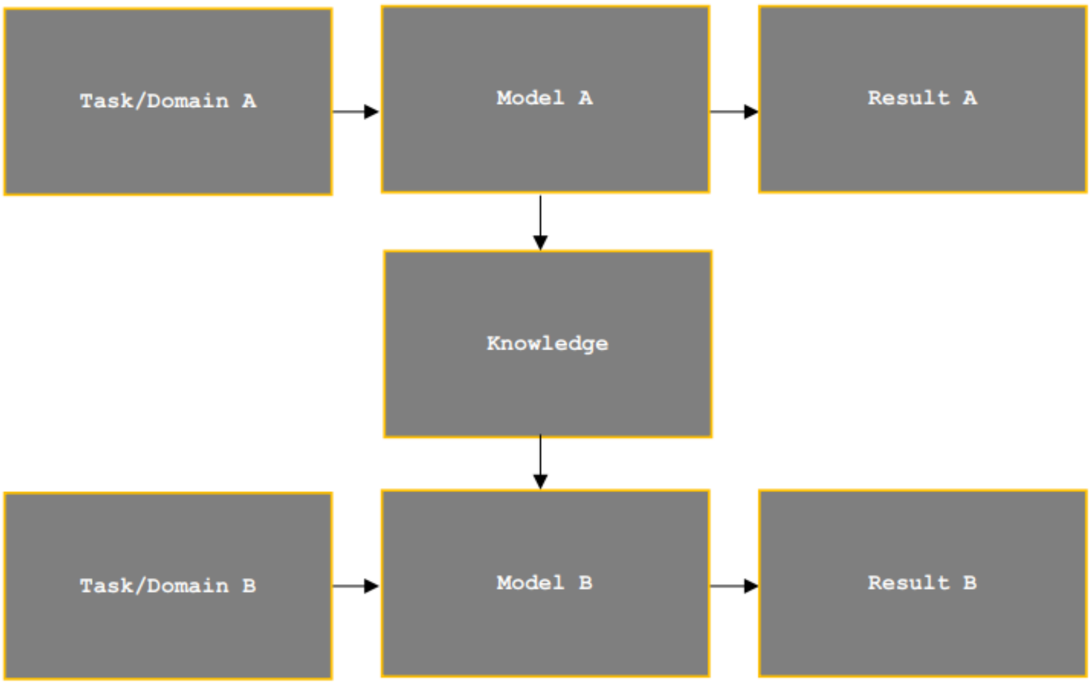


Figure 3. Information is passed through between models in transfer learning. Reprinted from "Learn TensorFlow 2.0: Implement Machine Learning and Deep Learning Models with Python" (p.96), by A. Manure & P. Singh, 2020, New York City, USA: Springer Professional.

As can be seen in figure 3, the transfer learning process involves sharing knowledge between models. Moreover, transfer learning has been adapted to the CNN architecture to produce numerous CNN variants like AlexNet, VGG16, GoogLeNet, and ResNet. These algorithms were the output of a Google competition called ImageNet, in which the models were trained with a large dataset of over 1.2 million images and approximately 1000 categories. The classification performance of these algorithms against classical computer vision is presented in figure 4.

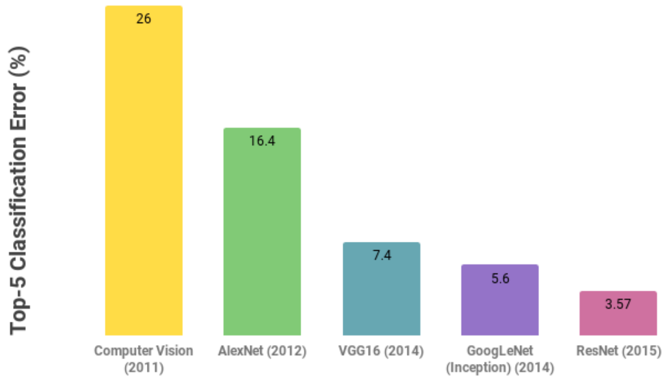


Figure 4. Transfer learning outperforms computer vision methodology. Reprinted from "Timeline of Transfer Learning Models" by Wittmann, 2019 (https://medium.com/analytics-vidhya/timeline-of-transfer-learning-models-db2a0be39b37)

In figure 4, the transfer learning CNN algorithms are more effective at the classification task than classical computer vision methodology. This motivated adapting these pre-trained transfer learning algorithms to our task of identifying distracted drivers. We therefore implemented the VGG16 (Error = 7.4%) and ResNet (Error = 3.75%) architectures as is described in the following section.

VGG16

As suggested, VGG16 is a CNN that has been trained on the ImageNet Dataset and designed to enable transfer learning. VGG16 is so named because it has 16 weighted layers, as is illustrated in figure 5.

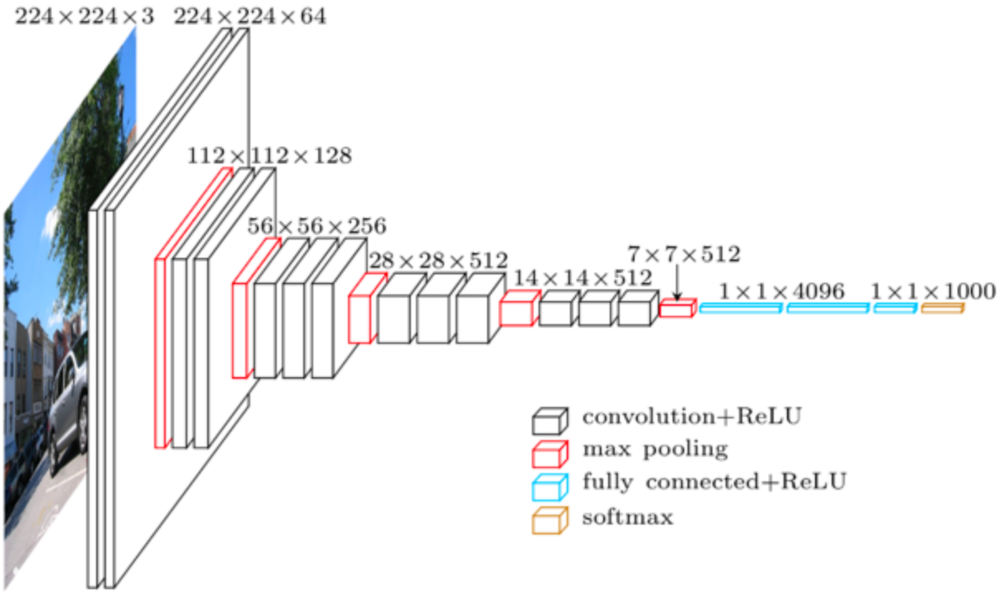


Figure 5. Information passing through the VGG16 model. Reprinted from "Very Deep Convolutional Networks for Large-Scale Image Recognition" by k. Simonyan & A. Zisserman, 2015.

The illustrated architecture in figure 5 represents a large network with approximately 138 million parameters (Simonyan & Zisserman, 2015). Moreover, VGG does not have many hyper-parameters in comparison with alternate CNN models. More precisely, it has a:

- convolution layer with a 3x3 filter, a stride of 1, while retaining the same padding
- maxpool layer with a 2x2 filter and a stride of 2

The 3x3 filter condenses a weighted combination of 9 pixels to a single pixel in the output layer and the stride hyperparameter determines the number of pixels the filter moves over the image in each step. Importantly, filter size affects the output volume of the model. The architecture of VGG16 (as depicted in figure 5) has a consistent set of convolution and maxpool layers throughout the entire architecture. The ultimate stage in the VGG16 network has 2 fully connected layers followed by a SoftMax for the output.

ResNet50

ResNet50 is another CNN transfer learning variant in which there are 48 Convolution layers along with a single MaxPool and a single Average Pool layer, as per figure 6.

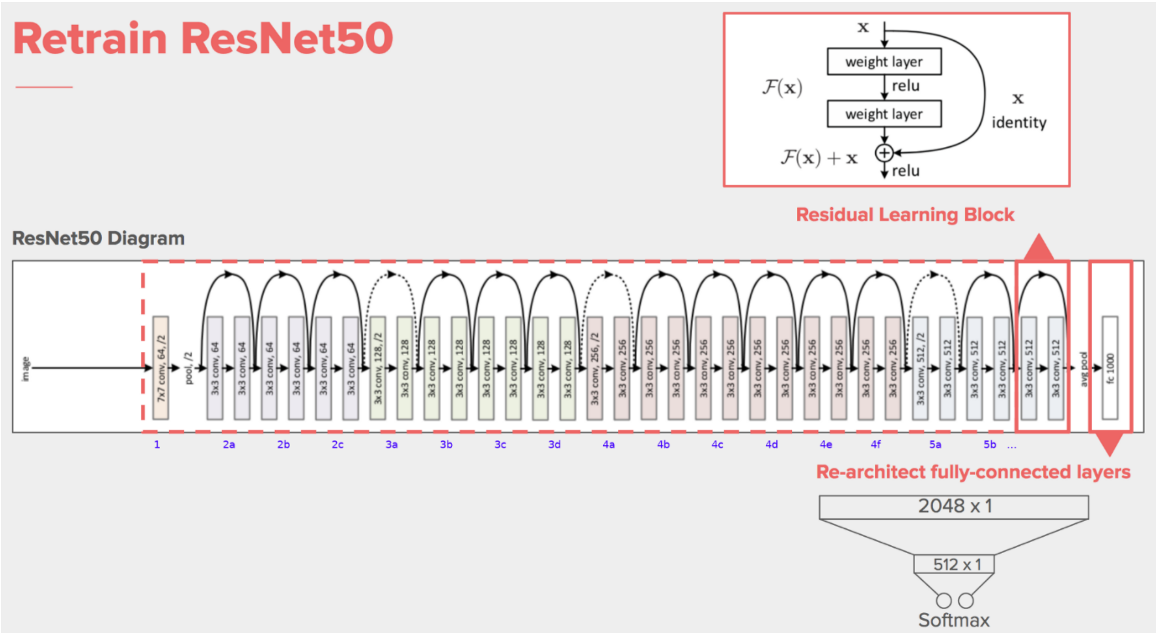


Figure 6. Information passing through the ResNet50 model. Reprinted from "Deep Residual Learning for Image Recognition" by K. He, X. Zhang, S. Ren, & J. Sun, 2015.

As illustrated in figure 6, the architecture of ResNet50 is distinctly different from VGG16. This algorithm is powerful because it enables deep networks (150+ layers) to train without encountering a notorious problem encountered by earlier researchers called "vanishing gradients". When earlier researchers had back-propagated the gradient through the network to earlier layers, the process of repeatedly multiplying values (with a magnitude of less than 1) generated gradients with extremely small values, which is computationally problematic. Consequently, the generated deep networks performance would plateau or even begin degrading rapidly.

ResNet negates the problem by introducing the process of skip connection, as is illustrated in figure 7.

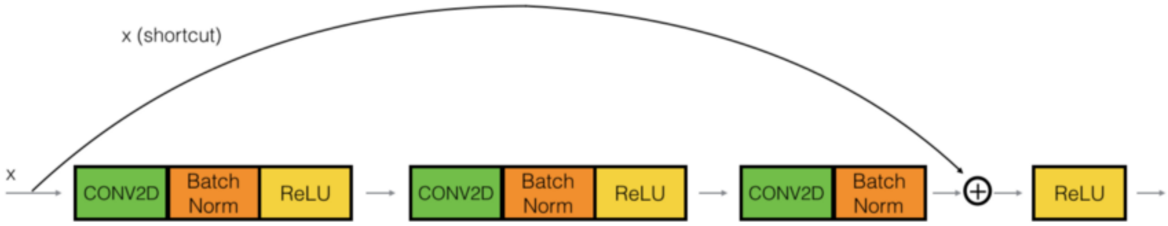


Figure 7. Skip connections enable information to bypass sections of the model. Adapted from "Convolutional Neural Networks (CNNs) Explained" by R. Balsys, 2021.

In figure 7, the central idea of skip connection is clearly illustrated as input to the model is provided directly to the output of the convolution layer. This architecture assists with preventing the 'vanishing gradient' problem.

Importantly, skip connection is applied before the RELU activation as empirical studies demonstrate that this is most effective. These skip connections mitigate the problem of vanishing gradient by creating this alternate shortcut path through which gradient information can pass. Additionally, they enable the model to learn an identity function which ensures that the higher layer will perform at least as well as lower layers, and not worse. (He et al. 2015)

Model Specification

The dataset consists of 22,000 labelled images which were grouped into two classes (i.e. focused and distracted). The dataset was very well balanced, with each class consisting of 11,000 images and we, therefore, decided to evaluate model accuracy with the metric outlined in equation 2:

$$Accuracy = \frac{TrueNegatives + TruePositive}{TruePositive + FalsePositive + TrueNegative + FalseNegative}$$

Equation 2. Accuracy metric for evaluation.

The images in the dataset depict 81 individuals doing various tasks and initially, these images in the focused and distracted sets were further split into 3 groups:

- 70% for model training
- 20% for model validation and hyper-parameter tuning
- 10% for testing the final accuracy of the model on unseen data

We used TensorFlow Keras to implement our model with the pre-trained weights of VGG16 and ResNet50 from the ImageNet dataset. Additionally, we used data augmentation as a regularization technique which helped to reduce overfitting by introducing slightly modified versions of existing images during the training phase. This effectively adjusts the shear range and zoom of the images by 20% so that the model gets exposed to multiple versions of the same image. Both transfer learning models considered (i.e. VGG16 and ResNet50) are individually connected to 2 final layers:

- A dense layer with 1024 nodes that have 'ReLU' (Rectified Linear Unit) activation
- An output layer with 2 nodes representing the 2 classes (i.e., distracted and focused drivers).

The output layer in both models uses SoftMax as an activation function, which is presented alongside ReLU in figure 8.

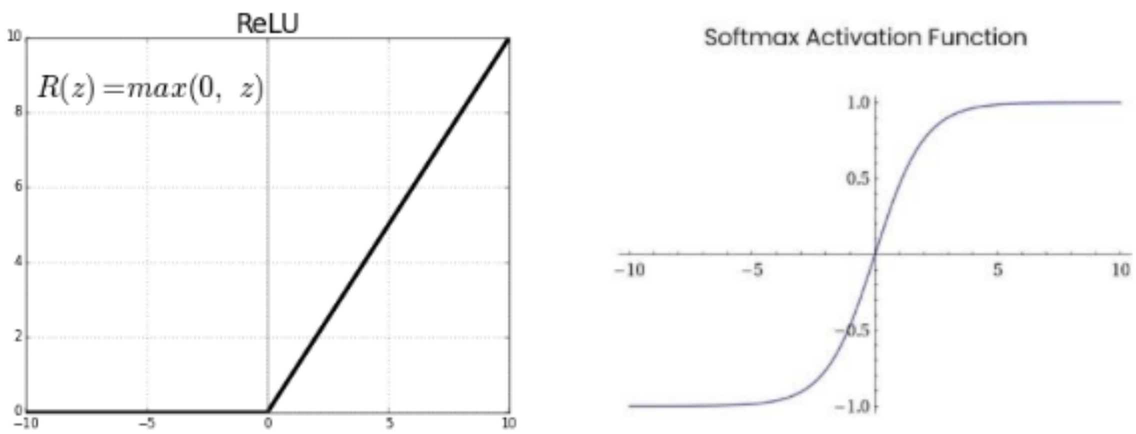


Figure 8. Comparison of ReLU and SoftMax activation functions.

As illustrated in figure 8, the two activation functions generate very different output values given the same input and this influences the way a model learns from data. On the one hand, ReLU is typically used in hidden layers as it lends itself to better computational performance and addresses the vanishing gradient problem. On the other hand, Softmax is well suited to be used in the last output layer of a CNN.

Model Output

Both the VGG16 and ResNet50 models performed well. When images are processed by either model, they are first reshaped to (224, 224) at the input stage, as shown in figure 9.

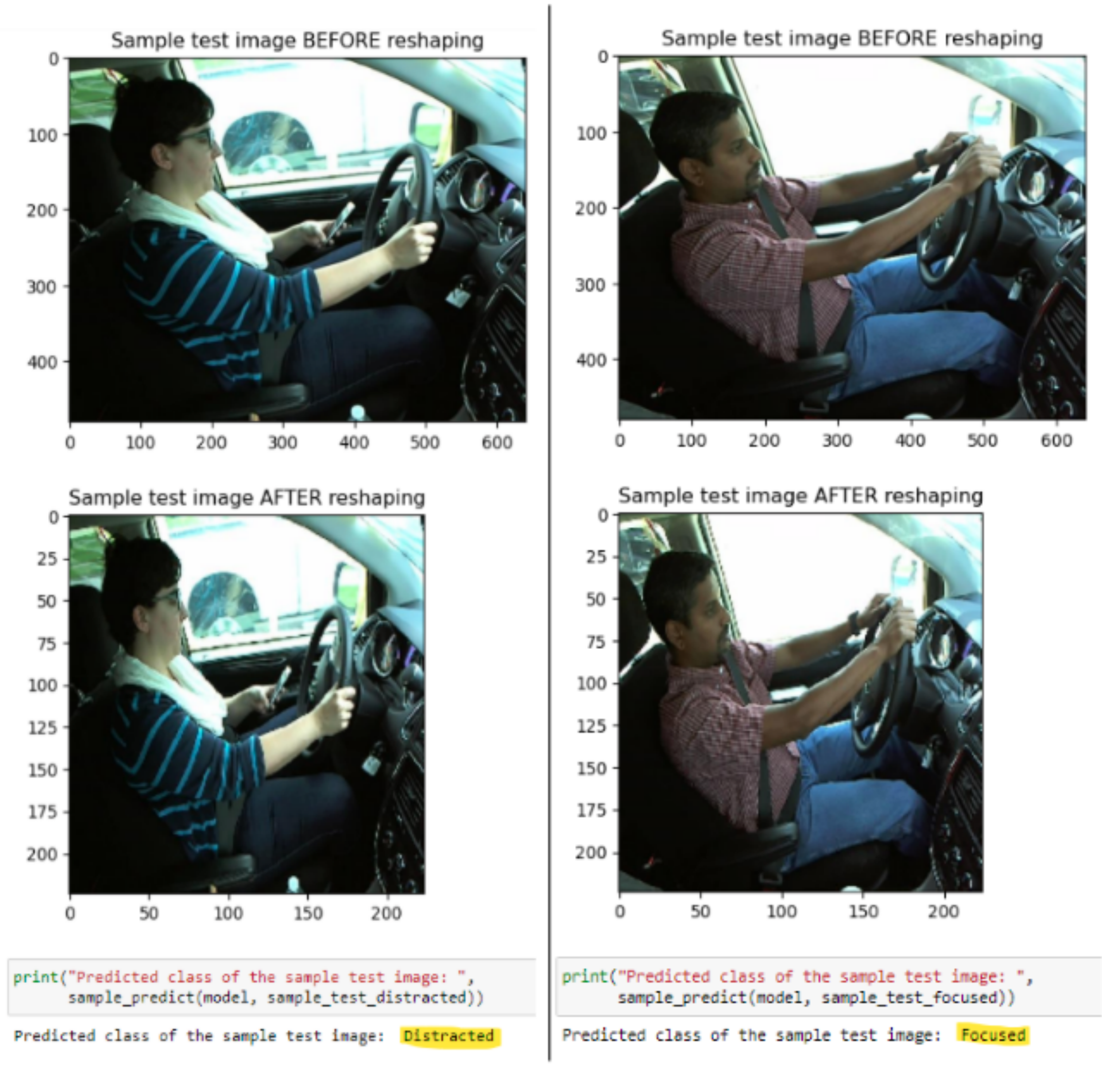


Figure 9. Images classified successfully by transfer learning (VGG16 left, ResNet15 right).

In figure 9, both images were correctly processed by VGG16 and ResNet50 models. The models correctly identify that the driver on the left is engaged in distracting behavior (i.e. texting on the phone while driving), and the driver on the right appears to be focused. These

The accuracy of ResNet50 was 73.93%, and for VGG16 it was surprisingly high at 99.33%, as shown in figure 10.

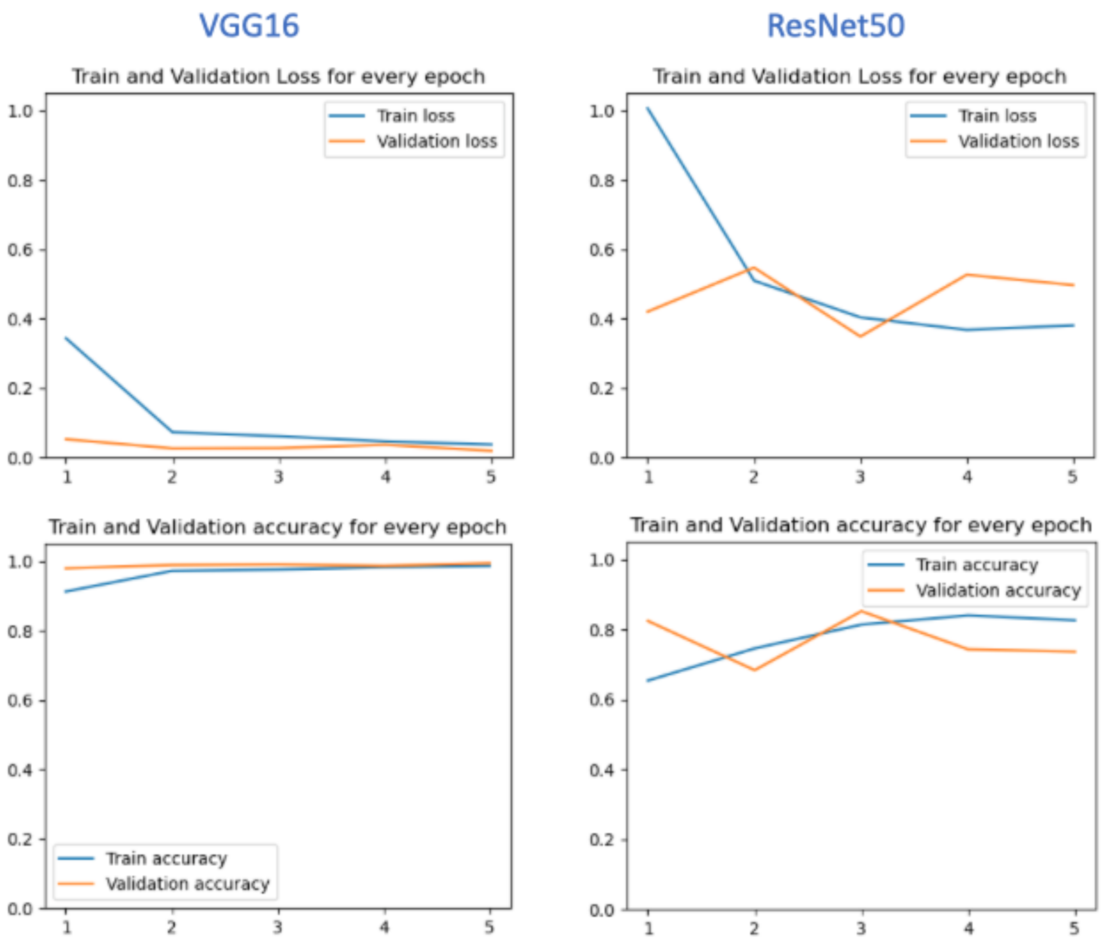


Figure 10. Both models performed well. Too well.

The plots illustrated in figure 10 show the relationship between loss and accuracy concerning epochs. Our model ran for a total of five epochs and the performance of VGG16 and ResNet50 are displayed on the left and right-hand sides of the figure accordingly.

The train and loss plot for VGG16 displays that the loss for both the train and validation sets seems to converge with decreasing loss as epochs increase. Furthermore, the orange line (validation loss) is consistently below the blue line (training loss) and these are both characteristics of a good model.

The ResNet model on the right-hand side, the plots look decent. As we can see from the chart on the bottom right, the model would perform better with two fewer epochs in which case the validation loss would be kept lower than the training loss. This is because when the validation loss crosses train loss, it indicates that the model is overfitting and that it will have a high variance when processing unseen data. While the validation set does not necessarily represent unseen test data perfectly, it indicates how the model may perform.

Unusually High Accuracy Scores

These high accuracy scores were a direct result of data leakage. As the dataset consists of 22'000 images of only 81 individuals, photos of the same individual dressed in the same clothes were almost certain to appear in all three datasets. This meant that much of the validation and test sets consisted of very similar data to the training set, as shown in figure 11.

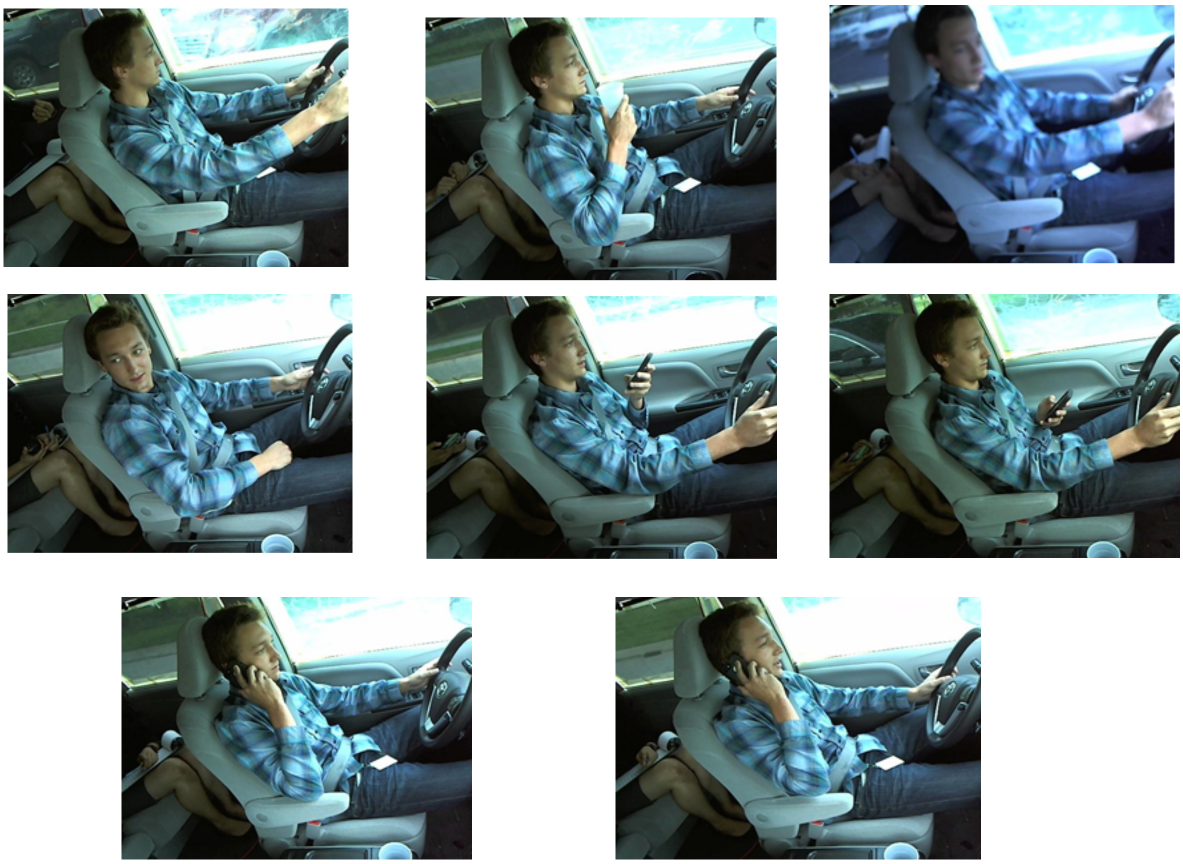


Figure 11. Very similar images of the same person within the dataset.

The images illustrated in figure 11 are a sample of the nearly 300 images of this particular individual in the dataset. These images are highly similar and when randomly split, there is a high likelihood of similar data being present in all sets.

To rectify this issue, the training, test, and validation sets were re-sampled based on individuals (id numbers) to achieve a similar 70%, 15%, and 15% split, and the models were subsequently retrained. **The new test scores were around 50% accuracy for both models (VGG16 and ResNet50), which was disappointing as this is roughly equivalent to random guessing.** In an attempt to regain the lost performance metrics, a further 5 fully connected layers and a 30% dropout layer was added. This still resulted in both transfer learning models reporting accuracy at no better than random guessing.

Building a CNN to Train from Scratch (No Transfer Learning)

As the dataset appeared to be inadequate for training the stock transfer learning models, we loosely utilized the principles of a VGG architecture as a basis for a new network as defined in table 1.

Layers	Parameters
2D Convolutional + Batch Normalisation	32 - Relu Activation
2D Convolutional + Batch Normalisation	32 - Relu Activation
Max Pooling	2x2
2D Convolutional + Batch Normalisation	64 - Relu Activation
2D Convolutional + Batch Normalisation	64 - Relu Activation
Max Pooling	2x2
2D Convolutional + Batch Normalisation	128 - Relu Activation
2D Convolutional + Batch Normalisation	128 - Relu Activation
Max Pooling	2x2
2D Convolutional + Batch Normalisation	256 - Relu Activation
2D Convolutional + Batch Normalisation	256 - Relu Activation
Max Pooling	2x2
Dropout	0.4
Flatten	
Dense	128 - Relu Activation
Dense	2 - Softmax Activation

Table 1. Manually constructed CNN architecture.

The architecture outlined in table 1 performed strongly in the 2014 ImageNet Challenge, as described by Simonyan & Zisserman (2015) which motivated its use for our project. Additionally, we added batch normalization after each convolution layer to assist with the generalisability of the model, and we added dense layers to finally produce a prediction. The model also utilizes ADAM optimization and categorical cross-entropy for its loss function. This model configuration generated promising results, as shown in figure 12.

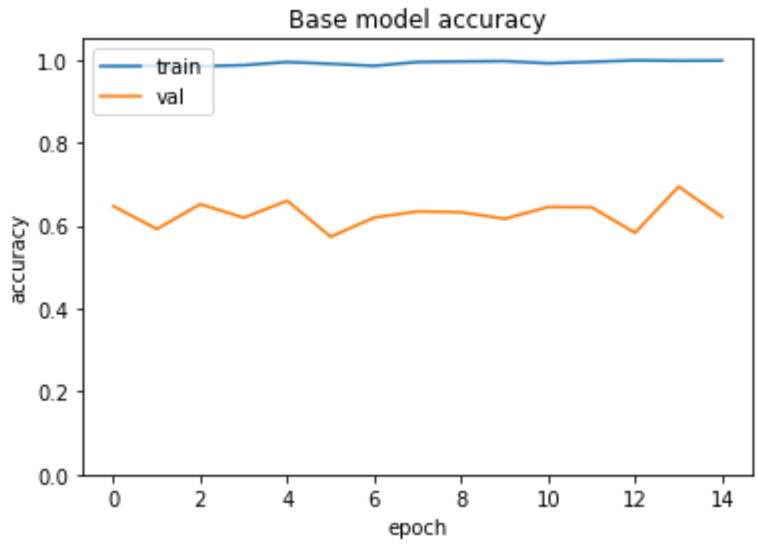


Figure 12. The manually constructed CNN presented promising accuracy.

Based on the learnings from developing the transfer learning model, we implemented Callbacks to select a model checkpoint based on the epoch that produced the maximum accuracy based on the validation set. Interestingly, the model performance was fairly stable throughout all epochs the Callbacks would not be expected to make any significant improvement of the model when it comes to processing unseen data.

In a further bid to increase generalizability, the model was further trained with augmented data. The images in the dataset were augmented with up to a rotation shift of 20 degrees, 20% width and height shifts, and 20% zoom, and once again a model checkpoint was selected based on the best validation accuracy, which in this case also happened to be the last epoch to be trained, as shown in figure 13.

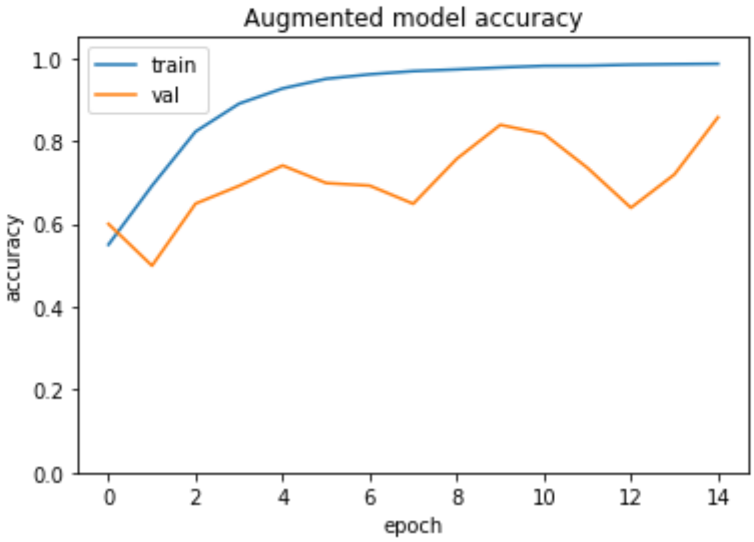


Figure 13. Model performance has substantially improved by using data augmentation.

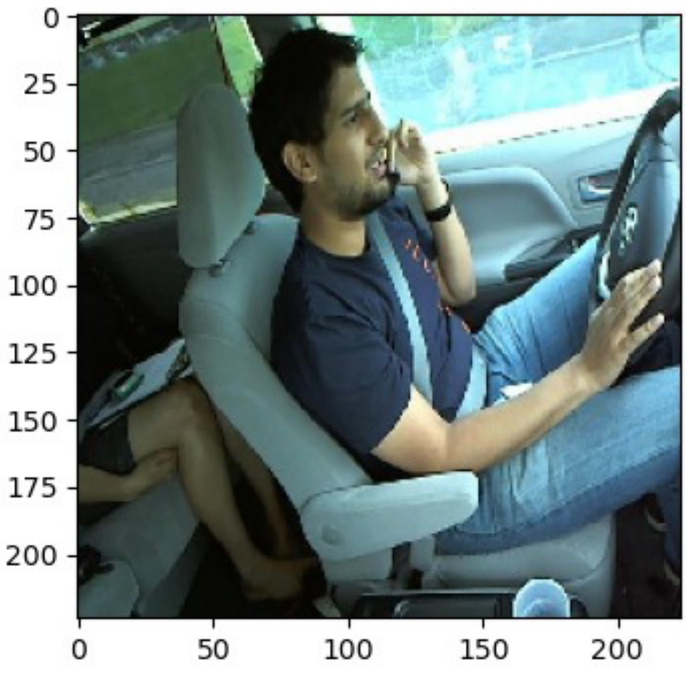
The accuracy scores corresponding to the model presented in figure 13 indicate that **model performance ultimately improved from 49.1% to 62.17% on the test set following the inclusion of the augmented dataset and validation accuracy settled on 85.71%.**

Limitations

The modeling aspect of this study was held back by a single limitation: access to adequate training data. There are 3 distinct ways that this problem manifested itself.

First of all, as all 22’000 images in the training dataset depict only 81 individuals, it is essential to control for data leakage between the test, train, and validation sets. Once the dataset was split according to person ID, each subset contained images of only a few individuals, and this meant that the CNNs were necessarily trained with very homogeneous data. We found other datasets, but they were unlabelled and without access to a trained model to label the data, there was no means by which we could resolve this issue promptly.

Second, there is only a single image for each instance. This is problematic because, in practice, a sequence of image frames is extracted from video recordings which can assist the modeling process. Alternatively, some roadside driver monitoring cameras are configured so that footage is recorded from multiple angles to avoid miss classifications of the type depicted in figure 14.




```
print("Predicted class of the sample test image: ",
Predicted class of the sample test image:  Focused
```

Figure 14. A distracted driver is classified as having been focused.

In figure 14, the CNN most likely misclassified the behavior as the individual's mobile phone is concealed behind his face, and it may appear as though he was adjusting his hair or scratching (actions which are classified as focused).

Third, all the images in this dataset are taken from the side which limits the ability of models to classify images taken from other angles. While image augmentation techniques can be used to invert images so that they can be interpreted from different orientations (i.e. for left-hand drive cars), they cannot be used to give the model exposure to photos taken from other angles (i.e. overhead).

There may be further ethical and political considerations that halt the widespread implementation of effective driver monitoring equipment. Despite the safety advantages and the potential to enhance driver compliance, some members of the public are concerned that implementing this technology on Queensland roads is Orwellian and serves only to enhance the power imbalance between governmental organizations and free citizens. This dilemma raises ethical questions regarding a person’s willingness to be photographed and have their image subject to such granular analysis. What right does the state have to initiate this analysis without any consent from the individual photographed?

As CNNs can autonomously classify images offline without the need for human intervention, implementation schemes of this technology can conceivably address some privacy concerns without sacrificing the safety benefits offered by driver monitoring technology. However, while this is technically possible, the process can easily be lost in translation when described to a non-specialist audience.

Conclusion

The topic of autonomous distracted driver detection/prevention is far-reaching, and it has silently developed alongside the technology that we rely upon in modern society. Governments worldwide have recognized the urgency with which this problem needs resolution and our role as aspiring data scientists is to assist their efforts and develop models that can autonomously and accurately identify these infringements of the law.

Through the trial and error of adapting various deep learning approaches, we constructed a CNN model with 62.17% test set accuracy and 85.71% validation set accuracy. While the results did not meet the initial expectations from the beginning of the project, the model performance seems to have been limited only by our access to suitable training data. Importantly, this experience highlighted how easily data leaks between datasets and the importance of preventing the issue of data leakage when addressing tasks of this nature.

In this research, we manually built a CNN and adapted two powerful transfer learning algorithms (i.e. VGG16 and ResNet50) to handle the classification problem. In doing so, we needed to make many decisions about hyperparameters and neural architectures (i.e. the number of dropout layers and dropout rates) and it was infeasible to investigate the impact of each conceivable combination of these decisions. Therefore, it is likely that the present models are suboptimal and can be improved (to a degree) without introducing new data.

For instance, Stochastic Gradient Descent (SGD) could be used in place of the ADAM Optimizer (with an appropriate increase in epochs). Moreover, there are entirely different CNN architectures and algorithms that could readily be adapted to this task (such as MobileNet or GoogleLenet) and they may provide superior performance. An alternate approach entirely would be to generate an ensemble of Neural Network classifiers and implement a voting system to determine the prediction of an overarching classifier.

For these reasons, we argue that our early efforts serve as a proof of concept and indicate the potential for using deep learning as a tool to automate the detection of distracted drivers.

References

Convolutional Neural Networks (CNNs) Explained. (2021). Python Lessons. Retrieved from https://pylessons.com/Keras-ResNet-tutorial/

Department of Transport and Main Roads QLD, 2021, *Queensland Road Crash Weekly Report, Report No: 1235*, Data Extracted: 13 Sep 2021

He k., Zhang X., Ren S. & Sun J., (2015) Deep residual Learning for Image Recognition

Manure A. & Singh P. (2020) Learn TensorFlow 2.0: Implement Machine Learning and Deep Learning Models with Python. Pp 96, https://doi-org.ezproxy.library.uq.edu.au/10.1007/978-1-4842-5558-2_4

Medium, (2019) Timeline of Transfer Learning Models, Retrieved from: https://medium.com/analytics-vidhya/timeline-of-transfer-learning-models-db2a0be39b37

Monash University Accident Research Centre. (2003). *Driver distraction: A review of the literature.* (No. 206). Melbourne, Australia.

Papantoniou, P., Papadimitriou, E., & Yannis, G. (2017). Review of driving performance parameters critical for distracted driving research. *Transportation Research Procedia*, *25*, 1796-1805. DOI: 10.1016/j.trpro.2017.05.148

Pylessons (2019), Convolutional Neural Networks explained: Residual Networks, Retrieved from: https://pylessons.com/Keras-ResNet-tutorial/

Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition.

State Farm Distracted Driver Detection, Kaggle 2016, https://www.kaggle.com/c/state-farm-distracted-driver-detection/data

Stavrinos, D., Jones, J., Garner, A., Griffin, R., Franklin, C., & Ball, D. et al. (2013). Impact of distracted driving on safety and traffic flow. *Accident Analysis & Prevention*, *61*, 63-70. DOI: 10.1016/j.aap.2013.02.003

Transport Operations (Road Use Management) Act *1995.* Retrieved from https://www.legislation.qld.gov.au/view/pdf/inforce/2021-03-29/sl-2010-0206

Wilson, F., & Stimpson, J. (2010). Trends in Fatalities From Distracted Driving in the United States, 1999 to 2008. *American Journal Of Public Health*, *100*(11), 2213-2219. DOI: 10.2105/ajph.2009.187179

Young, K. & Regan, M. (2007). Driver distraction: A review of the literature. In: I.J. Faulks, M. Regan, M. Stevenson, J. Brown, A. Porter & J.D. Irwin (Eds.). Distracted driving. Sydney, NSW: Australasian College of Road Safety. Pages 379-405.